Vol.2014-NL-218 No.5
2014/9/1

疑似的曖昧性の無い遷移に基づく ボトムアップ型非交差依存構造解析アルゴリズム

東 藍[1,a)]　松本裕治[1,b)]

概要：遷移に基づく依存構造解析アルゴリズムは，解析速度が速いことと，解析履歴を各段階の素性として利用できる利点を理由に広く用いられている．この種のアルゴリズムに対して，健全性，完全性，疑似的曖昧性の有無などの興味ある性質を論じることができる．しかし，それらの性質を具備していることを厳密に証明する，あるいはそれらの性質を具備したアルゴリズムを新たに考案する，などとった作業には系統的なあり方があるわけではなく，試行錯誤に頼ることになる．本稿では，遷移に基づく依存構造解析アルゴリズムに関するこのような作業過程において，組み合わせ論的議論が有効に活用できる可能性があることを示す．また，実際に組み合わせ論的議論を援用した着想に基づいて，健全かつ完全かつ疑似的曖昧性の無い遷移に基づく非交差依存構造解析アルゴリズムを考案する例を示す.

1. はじめに

遷移に基づく依存構造解析アルゴリズム [7] は解析対象の文の長さに対して線型な計算複雑性を持つ shift-reduce 法に基づくものが主流であり，その解析速度に加え，解析履歴に基づいた表現力の高い素性を用いることができる利点を持つ．この種のアルゴリズムに対して，健全性，完全性 [7]，疑似的曖昧性 [2] の有無などの興味ある性質を議論することができる．しかしながら，与えられたアルゴリズムがそのような性質を備えていることを厳密に証明すること，あるいは望ましい性質を備えたアルゴリズムを具体的に構成すること，などの作業過程において系統的なあり方が確立されているわけではない．また，新たなアルゴリズムの提案においてどのような着想からそのアルゴリズムを考案するに至ったかという過程は一般に評価の対象とならないため，このような作業過程における着想の在り方は文献等の記録にも残りにくい．結果として，このような作業過程においては再現性の無い試行錯誤を繰り返すことになりうる．

本稿では，このような作業過程において組み合わせ論的議論が援用できることを可能性があることを示す．組み合わせ論的議論とは，最も素朴な形態では離散対象の数を数えることである．また，ある離散対象の集合と別の離散対象の集合との数が等しいこと，すなわち 1 対 1 対応の存在を示すこと，さらにはその 1 対 1 対応の具体的な構成を示すこと，なども組み合わせ論的議論となる．

遷移に基づく依存構造解析アルゴリズムにおいては主に 2 種類の離散対象の集合が考えられる．アルゴリズムにおける可能な遷移列全体からなる集合，そして対象とする依存構造解析木のクラスである．これらに対する組み合わせ論的議論はアルゴリズムに関する諸性質の証明や議論に援用できる可能性がある．例えば，遷移に基づく依存構造解析アルゴリズムにおける可能な遷移列全体と対象とする依存構造木のクラスとの間に 1 対 1 対応があれば，アルゴリズムの健全性と完全性を示すとただちに疑似的曖昧性が無いことが導かれる，などといったことが想定される．

ここで，組み合わせ論的議論そのものが難しい，つまり離散対象の集合の要素数を閉じた式で表す，異なる 2 つの離散対象の集合間の要素数が等しく 1 対 1 対応が存在する，などといったことの着想，考察自体が試行錯誤を繰り返す再現性の乏しい作業過程となる，という指摘がありえるだろう．しかしながら，オンラインの大規模整数列辞典である The On-Line Encyclopedia of Integer Sequences (OEIS)[9] を用いればこの部分の困難を大幅に緩和できることが期待される．例えば，興味ある離散対象の集合 S が自然数 n を大きさのパラメタとして持っていれば，$n=0$ のときの $|S|$, $n=1$ のときの $|S|$，……を順に適当な方法で数え上げ，それを数列として OEIS にクエリとして投げれば $|S|$ の閉じた形式に関する予想を容易に得ることができる場合がある．たとえば，長さ n の文に対する非交差依存構造木の総数 a_n の最初の数項を $a_1=1$，$a_2=2$,

1　奈良先端科学技術大学院大学
　8916-5 Takayama, Ikoma, Nara 630-0192, Japan
a)　ai-a@is.naist.jp
b)　matsu@is.naist.jp

$a_3 = 7$，$a_4 = 30$，$a_5 = 143$，$a_6 = 728$，$a_7 = 3876$ などと数え上げることは，簡易なプログラムなどによって容易に達成できる．これを整数列とみなしたクエリで OEIS を検索すれば，この整数列がただちに OEIS 上に記録されている系列 A006013 から $a_n = \binom{3n+1}{n} \cdot \frac{1}{n+1}$ という a_n に関する閉じた式の予想が得られる．また，系列 A006013 に併記されている記述や参考文献から，非交差依存構造木の集合と数が等しいと予想される，すなわち 1 対 1 対応を持つと予想される集合が複数判明する．

本稿では，依存構造および遷移に基づく依存構造解析アルゴリズムに対する上記のような組み合わせ論的議論に触れつつ，望ましい性質を持つアルゴリズムを考案する際の着想や，アルゴリズムに対する各性質の厳密な証明に援用できることを具体的な例を通して示していく．本稿の以降の構成は以下のとおりである．2 節では依存構造木の形式的定義と，いくつかの依存構造木のクラスについてその組み合わせ論的特徴について触れておく．3 節では遷移に基づく依存構造解析アルゴリズムの形式的枠組みである遷移システム (transition system) を導入する．また，遷移システムの諸性質を論じる際に組み合わせ論的議論がどのように援用できるのかを示す．4 節では，依存構造解析アルゴリズムを新たに考案する際の作業仮説を組み立てる際に，組み合わせ論的議論がどのように援用できるのかを示す．そして実際に，健全かつ完全であり疑似的曖昧性の無い遷移に基づく依存構造解析アルゴリズムを，組み合わせ論的議論に基づいて組み立てた作業仮説に基づいて考案する．残念ながら，本節で考案するアルゴリズムは結果的に既存のアルゴリズム [3] と同等なものとなり新規性に欠けるが，組み合わせ論的な視点からいくつかの新たな考察を得ることを示す．

2. 依存構造木と組み合わせ論

本節では依存構造木の定式化と，そのいくつかのクラスに対する組み合わせ論的特徴について述べる．

定義 1. $\mathbf{x} = (w_1, w_2, \ldots, w_n)$ を長さ n の文とする．$\mathbf{x}$ に対する**依存構造森** (**dependency forest**) とは，有向グラフ $G = (N, A)$ であり，以下の全ての条件を満たすものである：

- $N = \{1, 2, \ldots, n\}$ ．
- $A \subseteq N \times N$ ．
- 全ての頂点は高々 1 つの親を持つ．すなわち，$(\forall i, i', j \in N)\,((i,j) \in A \wedge (i',j) \in A \rightarrow i = i')$ ．
- G が非循環である．

さらに G が連結であるとき，G を**依存構造木** (**dependency tree**) と呼ぶ．

定義 2. 依存構造木 $G = (N, A)$ が**非交差** (**projective**) であるとは，辺 $(i, j) \in A$ が存在するとき，頂点 i からその辺に覆われた区間の任意の頂点への経路が存在すること，すなわち以下を満たすことである．

$$(\forall i, j \in N)$$
$$((i,j) \in A \rightarrow (\forall k \in [\min(i,j), \max(i,j)])\,(i \rightsquigarrow k)) \ .$$

ここで $i, j \in N$ に対する $i \rightsquigarrow j$ なる表記は i から j への長さ 0 以上の有向経路が存在することを意味するものとする．

$\mathbf{x} = (w_1, w_2, \ldots, w_n)$ を長さ n の文とし，以下，$\mathbf{x}$ に対する非交差依存構造木全体からなる集合を $\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}(\mathbf{x})$，$\mathbf{x}$ に対する非交差依存構造木のうち i 番目の単語 w_i を根（最上位の親；root node）とするもの全体なる集合を $\mathcal{D}^{\mathrm{PROJ}}_{[1,i,n]}(\mathbf{x})$，と各々表記する．

以下，文 $\mathbf{x} = (w_1, \ldots, w_n)$ に対して，最初の語 w_1 または最後の語 w_n を根とする非交差依存構造木全体からなる集合 $\mathcal{D}^{\mathrm{PROJ}}_{[1,1,n]}(\mathbf{x})$, $\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}(\mathbf{x})$ についてその組み合わせ論的な特徴について触れておく．この集合に着目する理由については 4 節で述べる．ある文に対する依存構造木は左右を反転しても依然としてその文に対する妥当な依存構造木であるので，$\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}(\mathbf{x})$ についてのみ考えれば良い．この集合に対して，いくつかの小さな n に対する要素の個数を適当な方法で数え上げることにより，この集合の要素の個数が [9] の系列 A001764 と一致するであろうという予想を簡単に得ることができる．また，この集合と個数が同じ，すなわち 1 対 1 対応を持つであろう様々な組み合わせ論的対象に関する情報を [9] の系列 A001764 の参考文献から獲得することもできる．そのうちの 1 つを取り上げて，$\left|\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}(\mathbf{x})\right|$ の閉じた式を厳密に証明しておく．

定義 3. n 個の頂点を円上に反時計回りに順番に 1 から n までの数字でラベル付けして配置する．このとき，頂点を互いに交わらないように $n-1$ 本の直線（無向辺）で連結となるように結ぶとき，これを n 個の頂点からなる**非交差木** (**noncrossing tree**)[8] と呼ぶ．

定義 3 で定義されるグラフは n 個の頂点を $n-1$ 個の辺で連結となるように結ぶものであるため，非交差木という名前が表すように木の構造を持つことに注意されたい．この非交差木は，自然言語処理の分野における**弱い非交差依存構造木** (**weakly projective dependency tree**)[5] という構造の，組み合わせ数学の分野における対応物である．ここで弱い非交差依存構造木とは，通常の非交差依存構造木に加えて，文の最上位の親をある辺がまたぐような依存構造木を含めたものを指す．たとえば長さ 3 の文に対して $N = \{1, 2, 3\}$,　$A = \{(2, 1), (1, 3)\}$ である依存構造木は，通常の非交差依存構造木の定義には含まれないが，弱い非交差依存構造木の定義に含まれる．非交差木 (noncrossing tree) のある頂点 i を根と定めて，非交差木の全ての無向辺を有向辺に置き換えると，i を根とする弱い非交差依存構造木が得られる．また，相異なる非交差木に対してこの置き換えを適用すると常に相異なる弱い非交差依存構造木

が得られる．ところで，最初の語もしくは最後の語を根とする弱い非交差依存構造木全体からなる集合は，最初の語もしくは最後の語を根とする通常の非交差依存構造木全体からなる集合と完全に一致する．というのも，最初の語もしくは最後の語をまたぐ辺はありえないからである．したがって，以下の定理で示すように，非交差木は最後または最初の語を根とする非交差依存構造木の対応物でもある．さらに，非交差木は組み合わせ数学の分野ですでに既知の結果があるため，それを非交差依存構造木の議論に持ち込むことができる．

定理 1. $\mathcal{G}_n^{\mathrm{NT}}$ を n 個の頂点からなる非交差木全体の集合とする．このとき，$\mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ と $\mathcal{G}_n^{\mathrm{NT}}$ との間に 1 対 1 対応が存在する．

Proof. 有向木 $D \in \mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ に対して，そのすべての有向辺を無向辺に置き換え無向木 $G = (V, E)$ を得る操作を考える．この操作により得られる G は，D における非交差条件から，n 個の頂点からなる非交差木の条件を満たすことが分かる．実際，ある D がこの操作により非交差木の条件に違反する木 G となるとすると，$i < \ell < j < m$ に対して $\{i, j\} \in E$ かつ $\{\ell, m\} \in E$ となるが，このとき，元の D において $(i, j) \in A \wedge (j, i) \in A$ かつ $(\ell, m) \in A \wedge (m, \ell)$ となり，i または j から ℓ への有向経路および ℓ または m から j への有向経路が存在することになる．これはどの場合を考えても D に有向閉路または異なる 2 点間に異なる有向経路が存在するのいずれかとなり矛盾する．また，どの相異なる 2 つの $D, D' \in \mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ を取ってもそれらが上記の操作で同じ $\mathcal{G}_n^{\mathrm{NT}}$ の要素となることはない．実際，そのような場合があるとすると， D あるいは D' は n を根とする根付き木でなくなり矛盾する．すなわちこの操作によって定義される写像は $\mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ から $\mathcal{G}_n^{\mathrm{NT}}$ への単射である．逆に，$G \in \mathcal{G}_n^{\mathrm{NT}}$ を取り，G の全ての無向辺を n が根となるように有向辺へと置き換える操作を考える．この操作の結果は一意に定まり，結果として得られる有向グラフは n を根とする非交差依存構造木である．実際，G に対する非交差木の条件が，結果として得られる有向グラフの依存構造木としての非交差条件を保証する．また，この操作によって定義される $\mathcal{G}_n^{\mathrm{NT}}$ から $\mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ への写像は単射である．結局，$\mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}$ と $\mathcal{G}_n^{\mathrm{NT}}$ との間に両方向の単射が構成できることが分かったので，この 2 つは 1 対 1 対応する． □

n 個の頂点からなる非交差木の総数 $\left|\mathcal{G}_n^{\mathrm{NT}}\right|$ については既知であり，$\left|\mathcal{G}_n^{\mathrm{NT}}\right| = \binom{3n-3}{n-1} \cdot \frac{1}{2n-1}$ である [8]．したがって，定理 1 からの自明な系として以下がただちに導かれる．

系 1.

$$\left|\mathcal{D}_{[1,n,n]}^{\mathrm{PROJ}}\right| = \binom{3n-3}{n-1} \cdot \frac{1}{2n-1} \ . \tag{1}$$

以上で得られた組み合わせ論的事実は，節 4 において，遷移に基づく依存構造解析アルゴリズムを新たに考案する際の着想の指針として，また遷移に基づく依存構造解析アルゴリズムに対する健全性，完全性，疑似的曖昧性の有無を証明するための道具立てとして活用できることを示す．

3. 遷移システムと組み合わせ論

本節では遷移に基づく依存構造解析アルゴリズムを形式的に取り扱うための枠組みである**遷移システム** (**transition system**)[7] と，遷移システムに関する諸性質の定義を導入する．さらに，遷移システムに関する議論において組み合わせ論がどのように活用できるかを示す．なお，遷移に基づく依存構造解析システムには，遷移システムの他に，複数の遷移を選択することが可能な場合にその中から 1 つを選択する**オラクル** (**oracle**) という構成要素もあるが，これは本節で議論する遷移システムとは完全に独立に議論できるものであるため，詳細な定義や議論などは [7] などに譲り，本稿では省略する．

依存構造解析に対する遷移システムとは，四つ組 $S = (C, T, c_s, C_t)$ である．ここで C は**構成** (**configuration**) の集合であり，遷移に基づく依存構造解析アルゴリズムにおいて使用されるデータ構造を形式化したものである．各構成は，少なくとも残りのノードのバッファ β と依存構造の辺の集合 A を含むものとする．T は遷移の集合であり，各遷移は C から C への（部分）関数である．c_s は初期化関数であり，与えられた文 $\mathbf{x} = (w_1, \ldots, w_n)$ をバッファ β が $[1, \ldots, n]$ であるような構成へ写像する．$C_t \subseteq C$ は終端構成の集合であり，アルゴリズムの終了条件を定義する．

$S = (C, T, c_s, C_t)$ を遷移システムとする．与えられた文 $\mathbf{x} = (w_1, \ldots, w_n)$ に対する S における**遷移列** (**transition sequence**) とは，遷移の列 $\mathbf{t} = \left(t_1, t_2, \ldots, t_{|\mathbf{t}|}\right)$, $t_i \in T \quad (i = 1, \ldots, |\mathbf{t}|)$ であり，初期構成から終端構成へと至るようなもの，すなわち以下の条件を全て満たすものである：

(1) t_1 が $c_s(\mathbf{x})$ に対して適用可能．
(2) t_i が $t_{i-1}(\cdots t_2(t_1(c_s(\mathbf{x})))\cdots)$ に適用可能 $(i = 2, \ldots, |\mathbf{t}|)$ ．
(3) $t_{|\mathbf{t}|}\left(t_{|\mathbf{t}|-1}(\cdots t_2(t_1(c_s(\mathbf{x})))\cdots)\right) \in C_t$ ．

また， S において $\mathbf{x}$ に対して可能な遷移列全体からなる集合を考えることができる．これを S における $\mathbf{x}$ に対する**遷移列集合**と呼び，$T_S(\mathbf{x})$ と表記することにする．

遷移システム $S = (C, T, c_s, C_t)$ においてある遷移列 $\mathbf{t}$ が存在するとき，その遷移列によって最終的に得られる有向グラフ，すなわち $\mathbf{t}$ によって最終的に得られる終端構成における有向グラフの辺集合 A が表すグラフ，を $G(\mathbf{t})$ と表記することにする．また，与えられた文 $\mathbf{x}$ に対して S が生成しうる有向グラフ全体からなる集合を $\mathcal{G}_S(\mathbf{x})$ と表記

する．すなわち $\mathcal{G}_S(\mathbf{x}) := \bigcup_{\mathbf{t}\in\mathcal{T}_S(\mathbf{x})} G(\mathbf{t})$ である．

以上の定義を用いて，遷移システム S が依存構造木のあるクラス $\mathcal{D}$ に対して，完全であること，健全であること，を定義することができる．ここで**依存構造木のクラス**とは，依存構造木全体からなる集合の部分集合であり，何らかの条件によって一意に定められるものである．具体的には，例えば「非交差依存構造木全体からなる集合」が依存構造木のクラスの例である．以下では，依存構造木のあるクラス $\mathcal{D}$ に関して，与えられた文 $\mathbf{x}$ に対して限定したものを $\mathcal{D}(\mathbf{x})$ と表記することにする．

定義 4. 遷移システム S と依存構造木のクラス $\mathcal{D}$ が与えらているとする．このとき， S が $\mathcal{D}$ に対して

- **健全** (**sound**) であるとは，任意の文 $\mathbf{x}$ に対して $\mathcal{G}_S(\mathbf{x}) \subseteq \mathcal{D}(\mathbf{x})$ となることである．
- **完全** (**complete**) であるとは，任意の文 $\mathbf{x}$ に対して $\mathcal{G}_S(\mathbf{x}) \supseteq \mathcal{D}(\mathbf{x})$ となることである．
- **疑似的曖昧性が無い**とは，任意の文 $\mathbf{x}$ に対して，$(\forall \mathbf{t}, \mathbf{t}' \in \mathcal{T}_S(\mathbf{x}))(\mathbf{t} \neq \mathbf{t}' \to G(\mathbf{t}) \neq G(\mathbf{t}'))$ となることである．

ここで，文 $\mathbf{x}$ に対して，遷移システム S は遷移列全体の集合 $\mathcal{T}_S(\mathbf{x})$ から依存構造木のあるクラス $\mathcal{D}(\mathbf{x})$ への写像の構成的定義とみなすことができることを指摘しておく．遷移システムをこのような写像として理解したとき，健全性とは写像の値域が $\mathcal{D}(\mathbf{x})$ であること，完全性とは写像として全射であること，曖昧性が無いことは写像として単射であること，をそれぞれ意味する．さらに，このような理解においては $\mathcal{T}_S(\mathbf{x})$ と $\mathcal{D}(\mathbf{x})$ の組み合わせ論的な特徴を明らかにしておくことが重要な意味を持ちうることも分かる．$\mathcal{T}_S(\mathbf{x})$ と $\mathcal{D}(\mathbf{x})$ との個数の大小などから健全性，完全性，疑似的曖昧性の有無などがただちに分かる場合がありうるからである．特に，$\mathcal{T}_S(\mathbf{x})$ と $\mathcal{D}(\mathbf{x})$ の要素の個数が等しい，すなわち 1 対 1 写像が存在することが既知であれば，健全性，完全性，疑似的曖昧性が無いこと，のいずれか 2 つが明らかになれば残りの 1 つが自明に導かれる．すなわち，以下が成り立つ．

定理 2. S を遷移システム，$\mathcal{D}$ を依存構造木のクラスとし，任意の文 $\mathbf{x}$ に対して $|\mathcal{T}_S(\mathbf{x})| = |\mathcal{D}(\mathbf{x})|$ とする．このとき：

- S が $\mathcal{D}$ に対して健全かつ完全ならば疑似的曖昧性が無い．
- S が $\mathcal{D}$ に対して完全かつ疑似的曖昧性が無いならば健全．
- S が $\mathcal{D}$ に対して疑似的曖昧性が無いかつ健全ならば完全．

以上から，遷移システムに対して，対象とする依存構造木のクラスと遷移列集合の組み合わせ論的特徴を明らかにすることで，遷移システムの諸性質の議論を行う方針の選択肢を増やすことができることが分かる．たとえば，$\mathcal{T}_S$ と $\mathcal{D}$ との間の 1 対 1 写像の存在，健全性，完全性，疑似的曖昧性の無いこと，のうち証明が最も困難な性質を他の 3 つの比較的簡単な証明で代えることができる．

4. 遷移に基づく依存構造解析アルゴリズムの考案と組み合わせ論

本節では, 非交差依存構造木のクラス $\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}$ に対する遷移に基づく依存構造解析アルゴリズムで，健全かつ完全であり疑似的曖昧性の無いものを具体的に構成する問題を考えていく．そして，そのような問題を考察する過程において組み合わせ論的議論を具体的に援用するあり方について述べる．

4.1 組み合わせ論的議論による作業仮説の構築

$\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}(\mathbf{x})$ に対する遷移に基づく依存構造解析アルゴリズムで，健全かつ完全であり疑似的曖昧性の無いものを考案するとき，以下の 2 つの条件を全て満たす組み合わせ論的対象を探すのは良い方針だと思われる．

- $\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}(\mathbf{x})$ と組み合わせ論的な関係があること．
- 何らかの系列の集合と 1 対 1 対応があること．

1 つ目の条件は当然の要求である．2 つ目の条件は，2 節で指摘したように，遷移に基づく依存構造解析アルゴリズムが遷移列集合と対象の依存構造木のクラスとの間の写像の構成的定義とみなせることに注意した結果として得られるものである．特に，健全かつ完全であり疑似的曖昧性の無い遷移システムは遷移列集合と対象の依存構造木のクラスとの間の 1 対 1 写像の構成的定義に他ならない．したがって，木構造の組み合わせ論的対象と列構造の組み合わせ論的対象で 1 対 1 対応を持つものがそのような遷移システムの構成に関わるのではないかという作業仮説は有効であると考えられる．

以上の作業仮説のもとに, 非交差依存構造木のクラスに対応する組み合わせ論的対象の整数列である（と予想される）[9] の系列 A006013 に関わりのある組み合わせ論的対象を徹底的に調べ上げ，結果，[9] の系列 A001764 がそのような組み合わせ論的対象の有力な候補と挙げられるにいたった．これは 2 節で触れた最初または最後の語を根とする非交差依存構造木のクラスに対応する整数列である．また，上に挙げた第 2 の条件のとおり，3-ラニー系列 (3-Raney sequence)[1] と呼ばれる系列の集合とも対応する．また，[9] の系列 A006013 は系列 A001764 の自己畳み込みであることが既知であるが，これは $\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}(\mathbf{x}) = \bigcup_{i=1}^{n} \mathcal{D}^{\mathrm{PROJ}}_{[1,i,n]}$ であり，さらに $\mathcal{D}^{\mathrm{PROJ}}_{[1,i,n]}$ が $\mathcal{D}^{\mathrm{PROJ}}_{[1,i,i]} \times \mathcal{D}^{\mathrm{PROJ}}_{[i,i,n]}$ と 1 対 1 対応を持つことと完全に対応する．

以上から，$\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}$ に対して健全かつ完全であり疑似曖昧性の無い遷移に基づく依存構造解析アルゴリズムで，その遷移列集合が 3-ラニー系列と 1 対 1 対応を持つようなも

構成

$$C := \{(\sigma, \beta, d, A) \mid \sigma \in \mathbb{N}^*, \beta \in \mathbb{N}^*, d \in \{\text{“L”}, \text{“R”}\}, A \in N \times N\}$$

初期化

$$c_s : (w_1, w_2, \ldots, w_n) \mapsto ([\,], [1, 2, \ldots, n], \text{“R”}, \phi) \tag{2}$$

遷移

$$T := \{\text{Shift}, \text{Flip}, \text{Reduce}\} \ , \tag{3}$$

ここで

- Shift : $(\sigma, i \mid \beta, \text{“R”}, A) \mapsto (\sigma \mid i, \beta, \text{“L”}, A)$,
- Flip : $(\sigma, \beta, \text{“L”}, A) \mapsto (\sigma, \beta, \text{“R”}, A)$,
- Reduce :
  $(\sigma \mid i \mid j, \beta, \text{“L”}, A) \mapsto (\sigma \mid j, \beta, \text{“L”}, A \cup \{(j, i)\})$,
  $(\sigma \mid i \mid j, \beta, \text{“R”}, A) \mapsto (\sigma \mid i, \beta, \text{“R”}, A \cup \{(i, j)\})$.

終端条件

$$C_t := \{([n], [\,], \text{“L”}, A)\} \tag{4}$$

図 1 $\mathcal{D}^{\text{PROJ}}_{[1,n,n]}$ Shift=Flip=Reduce 遷移システムの定義

のが存在するのではないかという作業仮説を得る．

4.2 Shift=Flip=Reduce 遷移システム

上に述べた作業仮説をもとに， shift-reduce 法に類似の遷移システムを構成できるか考察したところ，図 1 に定義されるアルゴリズムを構成できることが分かった．

残念ながら本遷移システムは新規性に欠けるものであることが構成後に判明した．本遷移システムは林らがすでに提案しているアルゴリズム [3] と完全に対応付けることが可能だからである．林らのアルゴリズムにおける定式化は文献 [4][6] による動的計画法のアイデアを組み合わせているため，それらを取り除くと，実際，林らのアルゴリズムにおける scan 遷移が本アルゴリズムにおける Flip 遷移に対応し，また林らのアルゴリズムにおけるスタックトップに対する * 記号の付加が本アルゴリズムにおける d の値に対応する．

以下では，林らが考察していない遷移列集合の組み合わせ論的議論について言及しておく．この遷移システムの，長さ n の文 $\mathbf{x} = (w_1, \ldots, w_n)$ に対する，遷移列および遷移列全体の集合 $\mathcal{T}_S(\mathbf{x})$ のいくつかの特徴についてより詳細に述べておく．ここで述べる特徴のいくつかは以下で本遷移システムの諸性質を証明する際に再度言及する．初期構成においてキューに n 個の要素が置かれること，Shift のみがキューから要素を取り除く遷移であり 1 回の適用毎に 1 つの要素を取り除くこと，そして終端構成においてキューが空であることが要求されること，以上からどの遷移列においても Shift がちょうど n 回出現することが分かる．次に，初期状態における d が “R” であること，Shift が $d =$ “R” のときにしか適用可能でなく，さらに d の値を “R” から “L” に変えること，Flip が $d =$ “L” のときにしか適用可能でなく，さらに d の値を “L” から “R” に変えること，Reduce が d の値を変えないこと，終端条件における d が “L” であること，以上に注意すると，どの遷移列においても Shift と Flip の 2 種類の遷移は互いに交代で出現することが分かり，さらにどの遷移列の先頭においても最初の Shift が現れるまでは Flip が現れず，どの遷移列の末尾においても最後の Shift が現れて以降に Flip は現れないことも分かる．また，Reduce はスタック σ のトップに 2 つの要素がある限り常に適用可能である．Shift のみがスタックに要素を加える遷移であり，1 回適用される毎に 1 つ要素が加えられること，Reduce のみがスタックから要素を取り除く遷移であり，1 回適用される毎に 1 つ要素が取り除かれること，以上から，任意の遷移列の任意の先頭部分列において Reduce の数は Shift の数よりも少ない．

以上の観察をまとめると，shift=flip=reduce 遷移システムにおいては遷移の並びが以下の条件で表されるときかつそのときに限り遷移列となる．ただし，簡単のため Shift, Flip, Reduce を各々 S, F, R と略し，R^i が i 回の連続する R の適用を表すものとする．

$$\text{SFSR}^{i_1}\text{FR}^{j_1}\text{SR}^{i_2}\text{FR}^{j_2}\text{S}\cdots\text{SR}^{i_{n-2}}\text{FR}^{j_{n-2}}\text{SR}^{i_{n-1}} \ . \tag{5}$$

ただし

$$i_k, j_k \in \mathbb{N}_0 \ , \ j_{n-1} = 0 \ , \tag{6}$$

$$\sum_{k=1}^{n-1} i_k + j_k = n - 1 \ , \tag{7}$$

$$\sum_{k=1}^{m} i_k + j_k \leq m \quad (\forall m \in \{1, \ldots, n-1\}) \ . \tag{8}$$

ここで，式 (5) から (8) の条件で表される遷移列全体の集合が 3-ラニー系列と呼ばれる組み合わせ論的対象と 1 対 1 対応を持つことを指摘しておく．なお，この指摘から導かれる組み合わせ論的事実は本遷移システムの完全性の証明に利用される．

定義 5. 長さ $3n-2$ $(n \in \mathbb{N}_+)$ の **3-ラニー系列** (**3-Raney sequence**)[1] とは $3n-2$ 個の 1 または -2 からなる列 $\mathbf{s} = (s_1, \ldots, s_{3n-2})$，$s_i \in \{1, -2\}$ $(i = 1, \ldots, 3n-2)$ であり，以下の条件を全て満たすものである．

$$\sum_{i=1}^{3n-2} s_i = 1 \ , \tag{9}$$

$$\sum_{i=1}^{k} s_i > 0 \qquad (\forall k \in \{1, \ldots, 3n-2\}) \ . \tag{10}$$

定理 3. $\mathbf{x} = (w_1, \ldots, w_n)$ を長さ n の文，$\mathcal{T}_S(\mathbf{x})$ を式 (5) から (8) によって定義される shift=flip=reduce 遷移システムの $\mathbf{x}$ に対する遷移列全体の集合，$\mathcal{R}_{3n-2}$ を長さ $3n-2$ の 3-ラニー系列全体の集合とする．このとき， $\mathcal{T}_S(\mathbf{x})$ と

$\mathcal{R}_{3n-2}$ との間に 1 対 1 対応が存在する．

Proof. $\mathbf{t}$ を shift=flip=reduce 遷移システムの遷移列とする．$\mathbf{t}$ に対して，Shift および Flip の出現を全て 1 に，Reduce の出現を全て -2 に置き換える操作を適用することを考える．この置き換えにより得られる 1 と -2 からなる長さ $3n-2$ の列は 3-ラニー系列の条件を満たす．実際，式 (7) で表される条件はこの置き換えにより式 (9) で表される条件となり，式 (8) で表される条件はこの置き換えにより式 (10) で表される条件となる．また，この置き換えによって定義される $\mathcal{T}_S(\mathbf{x})$ から $\mathcal{R}_{3n-2}$ への写像は単射である．というのも，この操作により Shift と Flip がともに 1 に置き換えられるが，この 2 種類の遷移の出現は Shift を最初と最後とする交互の出現に限られるためである．逆に，$\mathbf{r}$ を長さ $3n-2$ の 3-ラニー系列とするとき，$\mathbf{r}$ における最初の 1 の出現を Shift に置き換え，以降の 1 の出現を全て Flip と Shift とに交互に置き換え，$\mathbf{r}$ における 2 の出現を全て Reduce に置き換えることを考える．この置き換えによって得られる Shift, Flip, Reduce の列は $\mathcal{T}_S(\mathbf{x})$ の遷移列の必要十分条件を全て満足する．実際，式 (9) で表される条件はこの置き換えにより式 (7) で表される条件となり，式 (10) で表される条件はこの置き換えにより式 (8) で表される条件となる．また，この置き換えによって定義される $\mathcal{R}_{3n-2}$ から $\mathcal{T}_S(\mathbf{x})$ への写像は明らかに単射である．結局，$\mathcal{T}_S(\mathbf{x})$ から $\mathcal{R}_{3n-2}$ への単射，及び $\mathcal{R}_{3n-2}$ から $\mathcal{T}_S(\mathbf{x})$ への単射が構成できることが分かったので，定理の主張が証明できた． □

長さが $3n-2$ の 3-ラニー系列全体からなる集合 $\mathcal{R}_{3n-2}$ の要素の個数は既知であり，$|\mathcal{R}_{3n-2}| = \binom{3n-3}{n-1}\cdot\frac{1}{2n-1}$ である [1]．したがって，定理 3 からの自明な系として以下がただちに得られる．

系 2. S を shift=flip=reduce 遷移システム，$\mathbf{x} = (w_1,\ldots,w_n)$ を長さ n の文とする．このとき $|\mathcal{T}_S(\mathbf{x})| = \binom{3n-3}{n-1}\cdot\frac{1}{2n-1}$.

系 3. S を shift=flip=reduce 遷移システムとする．このとき，任意の文 $\mathbf{x}$ に対して，$\mathcal{T}_S(\mathbf{x})$ と $\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}(\mathbf{x})$ との間に 1 対 1 対応があり，その要素の個数が等しい．

定理 2 と系 3 から健全性，完全性，疑似的曖昧性が無いこと，の 3 つの性質のうちどれか 2 つを示せば残り 1 つの成立がただちに分かる．しかしながら，本遷移システムに対するこれらの性質の証明は対応する林らのアルゴリズムに対するそれらの証明 [2] と重複するため省略する．

4.3 非交差依存構造木に対する Shift=Flip=Reduce 遷移システム

上記では最後の語を根とする非交差依存構造木のクラス $\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}$ に対する遷移システムを考案したが，これを若干修正することにより一般の非交差依存構造木のクラス $\mathcal{D}^{\mathrm{PROJ}}_{[1,n]}$ に対する遷移システムに改変できることを示しておく．文 $\mathbf{x} = (w_1,\ldots,w_n)$ が与えられたとき，ダミーの語 w_{n+1} を文の最後に配置し，そのダミーの語を根とし，さらにそのダミーにただ 1 つの子だけが生成されるように制限することで，$\mathbf{x}$ に対する非交差依存構造木が得られる．したがって，先に定義した $\mathcal{D}^{\mathrm{PROJ}}_{[1,n,n]}$ に対する shift=flip=reduce 遷移システムは，最後の語を根とする依存構造木のクラスに対して完全，健全かつ単射であるため，shift=flip=reduce 遷移システムに対して最後の語の子がただ 1 つしか生成されないような制限を加えることで非交差依存構造木のクラスに対する完全，健全，かつ単射な遷移に基づく依存構造解析アルゴリズムが得られることになる．そして，そのような図 1 の定義を若干修正することでそのような制限を容易に実現することができる．すなわち，図 1 における初期化および Shift 遷移の定義を以下のように修正する．

$$
\begin{aligned}
&c_s : (w_1,\ldots,w_n) \mapsto ([\,],[1,2,\ldots,n,n+1],\text{“R”},\phi) \quad \text{Shift}:\\
&\quad (\sigma, i \mid j \mid \beta, \text{“R”}, A) \mapsto (\sigma \mid i, j \mid \beta, \text{“L”}, A) \ ,\\
&\quad ([i],[j],\text{“R”},A) \mapsto ([i,j],[\,],\text{“L”},A) \ ,
\end{aligned}
\tag{11}
$$

この修正により，最後の Shift の後に適用できる Reduce が正確に 1 回だけとなり，これは最後の語がただ 1 つの子を持つ根であるような非交差依存構造木のクラスに対する遷移システムとなる．

5. 結論と今後の課題

本稿では，遷移に基づく依存構造解析アルゴリズムに関して，その諸性質を厳密に証明する，新たな種類のアルゴリズムを考案する，などといった作業過程において，組み合わせ論的議論が有効に活用できる場面があることを具体的な例を通して示した．このような組み合わせ論的議論の援用は，特に様々な作業仮説の構築において威力を発揮することが期待される．本稿では非交差依存構造木のクラスに対する遷移に基づく依存構造解析アルゴリズムにのみ限定した議論であったが，交差を許す依存構造木のクラスや，動的計画法やグラフに基づく依存構造解析アルゴリズムに対しても，同様に組み合わせ論的議論を援用することにより，込み入った議論をより系統的なあり方で展開していけるものと期待できる．

本稿で示した，遷移システムや依存構造木のクラスに対する議論において組み合わせ論を援用するあり方は，単に上記のような作業過程の一部を系統化するだけにとどまらず，より興味深い可能性を示唆しているように思われる．すなわち，依存構造木を木構造とはまったく異なる数理的対象と同一視することにより，新たな視点，新たな数理的手法を依存構造木に関する議論に持ち込むことが期待できる．

今後は，依存構造解析というタスクに導入可能な組み合

わせ論的議論を徹底的に洗い出し，それらを元に本タスクに新たな視点，手法，概念等を取り込んでいきたいと考えている．

参考文献

[1] Graham, R. L., Knuth, D. E. and Patashnik, O.: *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 2nd edition (1994).

[2] Hayashi, K.: Techniques for Improving Transition-based Dependency Parsing, *Doctoral Dissertation* (2013).

[3] Hayashi, K., Kondo, S. and Matsumoto, Y.: Efficient Stacked Dependency Parsing by Forest Reranking., *TACL*, Vol. 1, pp. 139–150 (2013).

[4] Huang, L. and Sagae, K.: Dynamic Programming for Linear-time Incremental Parsing, *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Association for Computational Linguistics, pp. 1077–1086 (2010).

[5] Kuhlmann, M.: Dependency Structures and Lexicalized Grammars, *Doctoral Dissertation, Universität des Saarlandes*, Vol. 44 (2007).

[6] Kuhlmann, M., Gómez-Rodríguez, C. and Satta, G.: Dynamic Programming Algorithms for Transition-based Dependency Parsers, *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, Association for Computational Linguistics, pp. 673–682 (2011).

[7] Nivre, J.: Algorithms for Deterministic Incremental Dependency Parsing, *Computational Linguistics*, Vol. 34, No. 4, pp. 513–553 (2008).

[8] Noy, M.: Enumeration of Noncrossing Trees on a Circle, *Discrete Mathematics*, Vol. 180, No. 1, pp. 301–313 (1998).

[9] OEIS Foundation Inc.: The On-Line Encyclopedia of Integer Sequences (2014).